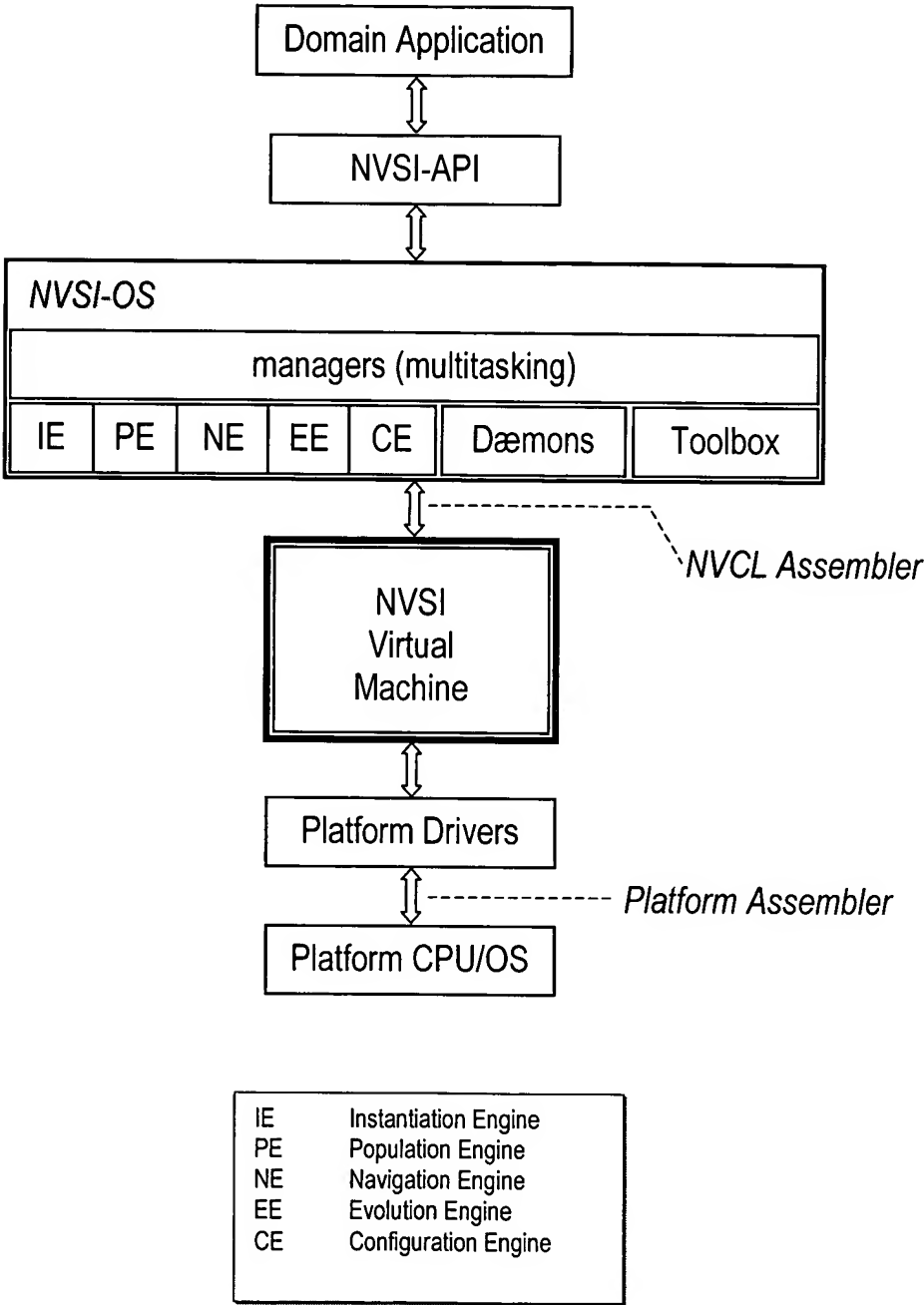


## **DRAWINGS**

**(see attached)**

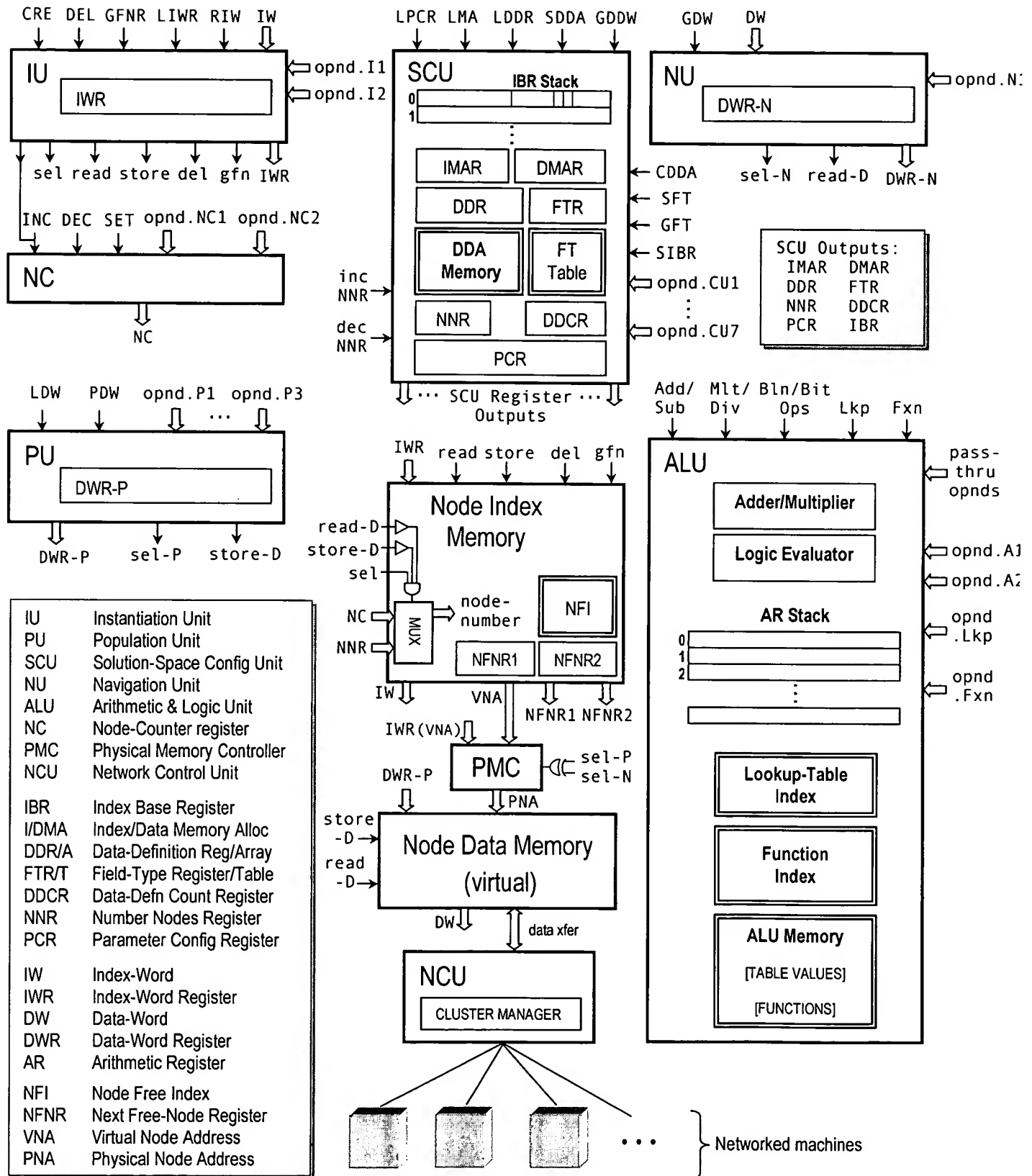
Drawing Sheet 1 / 7

FIG. 1. NVSI System Configuration Overview



## Drawing Sheet 2 / 7

FIG. 2. NVSI Virtual Machine Components and Logical Interconnections



## Drawing Sheet 3 / 7

FIG. 3. Pseudocode representation of the action of the Configuration Engine.

### Configuration Engine:

*Create & store m data definition words (DDW)*

**begin**

**for** *DDnum* = 1 **to** *m* **do**                   ; *m* is the number of different data-word definitions  
  ; in the domain solution-space (data memory).

- **create** a data definition word (DDW) in register (DDR),  
    according to parameters specified by Domain Application Program (DAP)
- **store** DDW into the DD array in Configuration Unit

**end do**

**end**

**end**

## Drawing Sheet 5 / 7

FIG. 5. Pseudocode representation of the action of the Population Engine.

**Population Engine:***Populate (evaluate & store) n nodes***begin**

**for** Node-Counter (NC) = 1 **to** *n* **do**      ; *n* is the number of nodes in the domain  
    ; solution-space (data memory) to be created.

- **read** IW from Index Memory at node-number (address) = NC
- **read** DDW pointed to by DDN in IW
- **evaluate** all fields in the Data Word according to the corresponding DDW
- **create** Data Word in DW register (DWR-P) in Population Unit

**if** Length-Data-Word **changed**

**then**

- memory manager **compute** new VNA
- **store** new VNA into corresponding IW  
    (and **update** VNA in IWR)

**else**

- **continue**

**end if**

- **store** DWR into Data Memory at address = VNA

**end do**

**end**

## Drawing Sheet 6 / 7

FIG. 6. Pseudocode representation of the action of the Navigation Engine.

**Navigation Engine:**

*Find and read a node data-word*

**begin**

**get** selected node-number from DAP

**read** IW from Index Memory at NC = node-number

**read** DW at corresponding VNA into DWR-N

**end**

## Drawing Sheet 7 / 7

FIG. 7. Pseudocode representation of the action of the Evolution Engine.

### Evolution Engine:

*Add, Delete or Modify a node*

**begin**

**get** selected node-number from DAP

**get** evolution *condition* (add new node, delete node, modify node) from DAP

**branch on condition**

- Add node:
  - **call** Instantiation procedure for NC = node-number
  - **call** Population procedure for same node,  
according to parameters specified by DAP
  - **exit**
- Delete node:
  - **call** Instantiation procedure (delete mode) for NC = node-number
  - **update** NFI and release space in data memory via PMC
  - **exit**
- Modify node:
  - **call** Navigation procedure for NC = node-number
  - **modify** fields in DWR-P as specified by DAP
  - **call** Population procedure for NC = node-number
  - **exit**

**end branch**

**end**